# **Attachment B**

Pages 49, 50, 54, and 103-109 of the Microfiche Appendix included in U.S. Patent App. Ser. No. 08/516,036

(11 pages)



Te 1165104

# THE BUILD STATES OF A MERRICAN

TO ALL TO WHOM THESE PRESENTS SHALL COME;

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

May 26, 2004

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF:

The Appendix Microfiche SERIAL NUMBER: 08/516,036 FILING DATE: August 16, 1995

By Authority of the COMMISSIONER OF PATENTS AND TRADEMARKS

M. K. HAWKINS Certifying Officer

# Instruction Set

All instructions are 32 bits in size, and use the high order 8 bits to specify a major operation code.

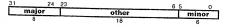


The major field is filled with a value specified by the following table:<sup>5</sup>

MAJOR	0	32	64	96	128	160	192	224
0	ERES	GSHUFFLEI	FMULADO 16	GMULADO1	LU16LAI	SAAS64LAI	EADDIO	BFE16
1	TSHUFFLE HAUX		FMULADO32	GMULADD2	LU16BAI	SAAS64BAL	EADDIUG	BENUE 16
2			FMULADD64	GMULADD4	LU16LI	SCAS54LAI	ESETIL	BENUGE 16
3	EMDEPI	GMDEP:		<b>GMULADD8</b>		SCAS64BAL	ESETIGE	BFNUL 16
4	EMUX	GMUX	FMULSUC16	GMULADD16	LU32LAI	SMAS64LAI	ESETIE	BFE32
5	EBMUX	G8MUX	FMULSUB32	GMULADD32	LU32BAI	SMAS64BAI	ESETINE	BENUE32
6	EGFMUL64	GGFMUL8	FMULSUB64	GMULADD64		SMUX64LAI	ESETIUL	BFNUGE32
7	CHEWAPON TOPE	SATHINGS MIN		GEXTRACT 128	LU32BI	SMUX64BAI	ESETHIGE	BFNUL32
8					LIGLAI	SIGLAI	ESUBIO	BFE64
9	ESW2ZLE	GSWIZZLE		GUMULADOS	L16BAI	SIGRAL	ESURIUO	BENUE64
10		63/920/ECOFY		GUMULADD4	L16LI	S1611	ESUBIL	BFNUGE 64
11		25w22(1)0mp		GUMUI.ADD8	LIGBI	S1681	ESUBIGE	BFNUL64
12	EDEPI	GDEPI	F.16	GUMULADD16	L32LAI	S32LAI	ESURIF	8FE128
13	EUDEPI	GUDEPI	F.32	GUMULADD32	L32BAI	S32BAI	ESUBINE	BENUE 128
14	EWTHI	GWTHI	F 64	GUMULADD64	L32L1	S32L1	ESUBIUL	BENUGE 128
15	EUWTHI	GUWTHI		GUEXTRAC1123	L328I	\$32BI	ESUBILICE	BFNUL 128
16			GFMULADD16		L64LAI	S64LAI	EADDI	BANDE
17			GFMULADD32	GEXTRACTI 16	L64BAI	S64BAI	EXORI	BANDNE
18			GFMULADO64	GEXTRACTI32	L64L1	S64LI	EORI	BL/SLZ
19			GENEY YOUTS!	GUEA" HACTIES	L64BI	S64BI	EANDI	BGE/BGEZ
20			GFMULSUB16	GEXTRACT	L I 28LAI	S128LAI	ESUBI	BE
21			GFMULSUB32	.1.64	L I 28BAI	S128BAI		BNF
22			GFMULSU866	GEXTRACT	L 128LI	S128L1	ENORI	BUL/BGZ
23			CFWULSUD:26	1 128	L 128BI	S128BI	ENANDI	BUGE/BLEZ
				G 1	L8I	Sål		BGATEI
25				G 2	LU8I			DOMILI
26	i			G 4				
27				G 8				
28		ECOPYI	GF. 16	G 16			ECOPYI	81
29			GF.32	G 32			200111	BLINKI
30			GF.64	G 64				172.171
31		E.MINOR	GF.128	G.128	LMINOR	SMINDR	E.MINOR	B.MINOR

major operation code field values

For the major operation field values A.MINOR, L.MINOR, E.MINOR, F.16, F.32, F.64, F.128, GF.16, GF.32, GF.64, G.1, G.2, G.4, G.8, G.16, G.32, G.64, S.MINOR, and B.MINOR, the lowest-order six bits in the instruction specify a minor operation code:



<sup>&</sup>lt;sup>5</sup>Blank table entries cause the Reserved Instruction exception to occur.

The minor field is filled with a value from one of the following tables:

E.MINOR	0	8	16	24	32	40	48	56
0	ÉADDO	ESUGO	EANDN		EADD	ESUB	ESPLIO	ESHRI
-	EADDUO	ESUBUO	EXOR		ESHLO	ESHLUO		
2	ESETL	ESUBL	EOR				ESHLIUO	EUSHRI
3	ESETGE	ESUBGE	EAND		ELMS	EULMS		
4	ESETE	ESUSE	EORN		EASUM	ESELECT8	ESHUFFLEI	ERORTI
5	ESETNE	ESUBNE	EXNOR		EROTL	ESHL		
6	ESETUL	SUBUL.	ENOR		ESHR	EUSHR	ESHLI	EMSHRI
7	ESETUGE	I ESUBUGE	ENAND I		EROTR	EMSHR		

## minor operation code field values for E.MINOR

F.size	0	8	16	24	32	40	48	56
0	FADD.N	FADD.1	FADD F	FAOD.C	FADD	FADD.X	FSEYE	FSETE X
1	FSUB N	FSUB.T	FSUB F	FSUBC	FSUB	FSUB.X	FSETNUE	<b>FSETNUE X</b>
2	FMUL N	FMUL T	FMUL F	FMUL.C	FMUL	FMUL.X	FSETNUGE	FSETNGE X
3	FDIV.N	FDIV.T	FDIV.F	FDIV C	FDIV	FOIV X	FSETNUL	FSETNUL X
	F UNARY.N	F.UNARY T	F.UNARY.F	F.UNARY C	F UNARY	F UNARY.Y		1
5								
6							-	T
7								_

#### minor operation code field values for F.size

GF.size	0	8	16	24	32	40	48	56
0	GFADD.N	CFADD.T	GFADD.F	GFADD.C	GFADD	GFADD.X	GESETE	GFSE1E X
1	GFSIJB N	GFSUB T	GFSUB.F	GFSUB C	GFSUB	GFSUB X	GFSETNUE	GESETNUE >
2	GFMUL.N	GFMUL.T	GFMUL F	GFMUL.C	GFMUL	GFMUL X	GESETNUGE	GFSETNGE >
3	GFDIV N	GFDIV.T	GFDIV F	GFDIV C	GFDIV	GFDIV.X	GESETNUL	GFSETNUL >
	GF UNARY.N	GF.UNARY.T	GF.UNARY F	GF.UNARY C	GF.UNARY	GF.UNARY X		
5								
6								

### minor operation code field values for GF.size

G size	0	8	16	24	32	40	48	56
0		GMUL	GANDN		GADD	GSUB	GEXPAND	GSHR
1		GUMUL	GXOR		GCOMPRESS	GUCOMPRESS		,
2	GSETL	GDIV	GOR				GUEXAPAND	GUSHR
3	GSETGE	GUDIV	GAND					1
4	GSETE	GSUB	GORN		GEXPAND	GUEXPAND	GCOMPRESSI	GROTE
5	GSETNE		GXNOR		GROTL	GSHL	<b>GUCOMPRESS</b>	
6	GSETUL		GNOR		GSHR	GUSHA	GSHII	GMSHR
7	GSETUGE		GNAND		GROTE	GMSHR		1

#### minor operation code field values for G.size

L-MINOR	0	8	16	24	32	40	48	56
0	LU16LA	L16LA	L64LA	£8		-		-
1	LU16BA	L16BA	L64BA	LUB				
2	LU16L	L16L	L64L					
3	LU16B	L16B	L64B					
4	LU32LA	L32LA	L128LA					
5	LU32BA	L32BA	L128BA					
- 6	LU32L	L32L	L128L					
7	LU32B	L32B	L128B					

minor operation code field values for L.MINOR

- 50

microunity

II I I

GUMULADD2, GUMULADD4, GUMULADD8, GUMULADD16, GUMULADD32 GMUX, GMUXGATHER, GSCATTERMUX, G.EXTRACT, 128: GroupTernary(major,size,ra,rb,rc,rd) G.EXTRACT.I, G.EXTRACT.I.64: GroupExtractImmediate(major,ra,rb,rc,minor) G.1, G.2, G.4, G.8, G.16, G.32:

case minor of

G.SHL, G.SHR, G.USHR, G.ADD, G.SUB, G.MUL, G.UMUL G.AND, G.OR, G.XOR, G.ANDN, G.NAND, G.NOR, G.XNOR, G.ORN, G.SET.E, G.SET.NE, G.SET.L, G.SET.GE, G.SET.UL, G.SET.UGE G.COPY, G.SWAP, G.DEAL, G.SHUFFLE, G.COMPRESS, G.EXPAND, G.GATHER, G.SCATTER:

Group(minor,major,ra,rb,rc) G.COMPRESS.I. G.EXPAND.I. G.SHL.I. G.SHR.I. G.U.SHR.I:

GroupShortImmediate(minor,major,ra,simm,rc) G.EXTRACT.I: GroupExtractImmediate(major.ra.rb.rc.minor)

others: raise ReservedInstruction

endcase GFMULADD16, GFMULADD32, GFMULADD64, GFMULSUB16, GFMULSUB32, GFMULSUB64: GroupFloatingPointTernary(major,ra,rb,rc,rd)

GF.16, GF.32, GF.64, GF.128: case minor of GF.ADD.N. GF.SUB.N. GF.MUL.N. GF.DIV.N.

GF.ADD.T, GF.SUB.T, GF.MUL.T, GF.DIV.T, GF.ADD.F. GF.SUB.F. GF.MUL.F. GF.DIV.F. GF.ADD.C. GF.SUB.C. GF.MUL.C. GF.DIV.C. GF.ADD. GF.SUB. GF.MUL. GF.DIV. GF.ADD.X, GF.SUB.X, GF.MUL.X, GF.DIV.X, GF.SET.E, GF.SET.NE, GF.SET.NE, GF.SET.NE, GF.SET.NUE, GF.SET.NUGE, GF.SET.UGE, GF.SET.UL, GF.SET.NUL, GF.SET.E.X, GF.SET.NE.X, GF.SET.UE.X, GF.SET.NUE.X, GF.SET.L.X, GF.SET.NL.X, GF.SET.NGE.X, GF.SET.GE.X:

GroupFloatingPoint(minor.op, major.size, minor.round, ra, rb, rc) GF.UNARY.N, GF.UNARY.T, GF.UNARY.F, GF.UNARY.C, GF.UNARY, GF.UNARY.X

case unary of GF.ASS, GF.NEG, GF.SQR, GF.HALF, GF.SINGLE, GF.DOUBLE, GF.QUAD,

GroupFloatingPointUnary(unary op, major size,

minor round, ra, rc) raise ReservedInstruction

endcase others raise ReservedInstruction

endcase L.MINOR

case minor of L16L, LU16L, L32L, LU32L, L64L, L128L, L8, LU8, L16LA, LU16LA, L32LA, LU32LA, L64LA, L128LA, L16B, LU16B, L32B, LU32B, L64B, L128B, L16BA, LU16BA, L32BA, LU32BA, L64BA, L128BA:

## Group

These instructions take two operands, perform a group of operations on partitions of bits in the operands, and catenate the results together.

#### Operation codes

0000000	
G.ADD.2	Group add pecks
G.ADD.4	Group add nibbles
G.ADD.8	Group add bytes
G.ADD.16	Group add doublets
G.ADD.32	Group add quadlets
G.ADD.64	Group add octlets
G.AND <sup>14</sup>	Group and
G.ANDN <sup>15</sup>	Group and not
G.COMPRESS.1	Group compress bits
G.COMPRESS.2	Group compress pecks
G.COMPRESS.4	Group compress nibbles
G.COMPRESS.8	Group compress bytes
G.COMPRESS.16	Group compress doublets
G,COMPRESS.32	Group compress quadlets
G.COMPRESS.64	Group compress octlets
G.DIV.64	Group signed divide octlets
G.EXPAND.1	Group signed expand bits
G.EXPAND.2	Group signed expand pecks
G.EXPAND.4	Group signed expand nibbles
G.EXPAND.8	Group signed expand bytes
G.EXPAND.16	Group signed expand doublets
G.EXPAND.32	Group signed expand quadlets
G.EXPAND.64	Group signed expand octlet
G.GATHER.2	Group gather pecks
G.GATHER.4	Group gather nibbles
G.GATHER.8	Group gather bytes
G GATHER.16	Group gather doublets
G.GATHER.32	Group gather quadlets
G.GATHER.64	Group gather octlets
G.GATHER. 12816	Group gather hexlets
G.MUL.117	Group signed multiply bits
G.MUL.2	Group signed multiply pecks
G.MUL.4	Group signed multiply nibbles
G.MUL.8	Group signed multiply bytes

- 103 -

<sup>&</sup>lt;sup>14</sup>G.AND does not require a size specification, and is encoded as G.AND.1.
<sup>15</sup>G.ANDN does not require a size specification, and is encoded as G.ANDN.1. G.ANDN is used as the encoding for G.SET.L.1, and by reversing the operands, for G.SET.UL.1.
<sup>16</sup>G.G.ATHER.128 is encoded as G.GATHER.1
<sup>17</sup>G.MU.L.1 is used as the encoding for G.DMULI.

G.MUL.32 Group signed multiply quadlets G.MUL.32 Group signed multiply octolets G.MUL.32 Group signed multiply octolets G.NAND18 Group nor G.NOR19 Group nor G.NOR19 Group or G.OR20 Group or G.OR21 Group or G.OR21 Group or G.OR21 Group or G.OR21 Group polynomial divide bits G.POLY.2 Group polynomial divide bits G.POLY.3 Group polynomial divide pecks G.POLY.4 Group polynomial divide doublets G.POLY.5 Group polynomial divide doublets G.POLY.6 Group polynomial divide doublets G.POLY.7 Group polynomial divide doublets G.POLY.8 Group polynomial divide doublets G.POLY.92 Group polynomial divide doublets G.POLY.16 Group polynomial divide doublets G.POLY.16 Group polynomial divide doublets G.POLY.17 Group polynomial divide doublets G.POLY.18 Group polynomial divide doublets G.POLY.19 Group polynomial divide doublets G.POLY.19 Group polynomial divide doublets G.POLY.10 Group rotate left pocks G.ROTL.4 Group rotate left nibbles G.ROTL.2 Group rotate left nibbles G.ROTL.2 Group rotate left nibbles G.ROTR.2 Group rotate right pocks G.ROTR.3 Group rotate right doublets G.ROTR.4 Group scatter right doublets G.ROTR.5 Group scatter right doublets G.SCATTER.4 Group scatter robuptes G.SCATTER.8 Group scatter r	IG.MUL. 16	Group signed multiply doublets
GMUL 64 Group signed multiply octlets GNAND19 Group nand GNOR19 Group nand GNOR19 Group nor GOR20 GNOR21 Group or GOR20 GNOR21 Group or GNOR21 GROUP or GNOR22 GROUP or GNOR22 GROUP or GNOR22 GROUP or GNOR23		Group signed multiply doublets
G.NANDIB Group nand G.NORIB Group nor G.OR20 Group or G.OR30 Group or G.OR30 Group or G.OR31 Group or not G.OR31 Group or not G.OR31 G.OR31 G.OR32 Group polynomial divide bits G.POLY.1 Group polynomial divide pecks G.POLY.4 Group polynomial divide inbloles G.POLY.8 G.POLY.8 G.POLY.8 G.POLY.8 G.POLY.8 G.POLY.9 G.POLY.8 G.POLY.9 G.POLY.		Croup signed multiply quadrets
GNORI® GORZ GORZ GORZ GORZ GORZ GORZ GORZ GORZ		
GORPA GONPA		
GORNI Group or not GPOLY Group or not GPOLY 1 Group polynomial divide bits GPOLY 2 Group polynomial divide pecks GPOLY 2 Group polynomial divide pecks GPOLY 8 Group polynomial divide pecks GPOLY 8 Group polynomial divide doublets GPOLY 9 Group polynomial divide doublets GPOLY 32 Group polynomial divide doublets GPOLY 4 Group polynomial divide octlets GROTL 9 Group rotate left pecks GROTL 4 Group rotate left doublets GROTL 6 GROTL 6 Group rotate left doublets GROTL 6 GROTL 7 Group rotate left doublets GROTL 7 GROTL 7 GROUP 10 GROTL 7 GROTL 7 GROUP 10 GROTL 7 GROT		
GPOLY.1 Group polynomial divide bits GPOLY.2 Group polynomial divide pecks GPOLY.4 Group polynomial divide pecks GPOLY.4 Group polynomial divide pecks GPOLY.8 GPOLY.16 Group polynomial divide oubbles GPOLY.16 Group polynomial divide doublets GPOLY.32 Group polynomial divide quadlets GPOLY.64 Group polynomial divide quadlets GPOLY.64 GROTL.2 Group polynomial divide quadlets GROTL.9 Group rotate left pecks GROTL.9 GROTL.9 Group rotate left nibbles GROTL.8 GROTL.9 Group rotate left doublets GROTL.9 GROTL.9 Group rotate left doublets GROTL.9 GROTL.9 Group rotate left oublets GROTL.9 GROTL.9 GROTL.9 GROUP rotate left oublets GROTT.9 GR		
GPOLY2 Group polynomial divide pecks GPOLY3 Group polynomial divide pecks GPOLY4 Group polynomial divide nibbles GPOLY5 GPOLY5 GPOLY7 GROUP polynomial divide doublets GPOLY7 GPOLY7 GROUP polynomial divide doublets GPOLY7 GPOLY7 GROUP polynomial divide doublets GPOLY7		
GPOLY 4 Group polynomial divide hibbles GPOLY 8 Group polynomial divide bytes GPOLY 16 Group polynomial divide bytes GPOLY 16 GPOLY 32 Group polynomial divide doublets GPOLY 32 Group polynomial divide quadlets GPOLY 34 Group polynomial divide quadlets GROTL 2 Group polynomial divide quadlets GROTL 34 Group polynomial divide quadlets GROTL 5 GROTL 6 GROTL 6 GROTL 8 Group rotate left pecks GROTL 8 GROTL 8 Group rotate left doublets GROTL 8 GROTL 9 Group rotate left doublets GROTL 9 G		
G.POLY.8 Group polynomial divide bytes G.POLY.16 Group polynomial divide doublets G.POLY.32 Group polynomial divide doublets G.POLY.64 Group polynomial divide quadlets G.POLY.64 Group polynomial divide octlets G.POLY.65 Group rotate left pecks G.POLL.2 Group rotate left pecks G.POLL.8 Group rotate left pibbles G.POLL.8 Group rotate left pecks G.POLL.8 Group rotate left pecks G.POLL.8 Group rotate left develotes G.POLL.9 Group rotate left develotes G.POLL.16 Group rotate left develotes G.POLL.16 Group rotate left develotes G.POLL.17 Group rotate left pecks G.POLL.18 Group rotate left pecks G.POLL.19 Group rotate right quadlets G.POLL.19 Group rotate right collets G.POLL.20 Group scatter rotates G.POLL.20 Group scatter rotates G.POLL.20 Group scatter pecks G.POLL.20 Group scatter pecks G.POLL.21 Group scatter politics G.POLL.21 Group scatter politics G.POLL.22 Group scatter politics G.POLL.23 Group scatter politics G.POLL.24 Group shift left politics G.POLL.25 Group shift left politics G.POLL.25 Group shift left politics G.POLL.25 Group shift left doublets		
GPOLY.16 GPOLY.16 GPOLY.22 Group polynomial divide doublets GPOLY.23 Group polynomial divide quadlets GPOLY.64 Group polynomial divide quadlets GPOLY.64 Group protate left packs GROTL.2 GROTL.4 Group rotate left mibbles GROTL.65 GROTL.66 GROTL.67 GROTL.7 GROUP rotate left doublets GROTL.7 GROTL.7 GROUP rotate left doublets GROTL.7 GROTL.7 GROUP rotate left quadlets GROTL.7 GROTL.		
GPOLY 32 Group polynomial divide quedlets GPOLY 64 Group polynomial divide octlets GROTL2 Group rotate left pecks GROTL3 GROTL3 GROUP rotate left pibbles GROTL4 Group rotate left pecks GROTL4 Group rotate left pecks GROTL5 GROTL5 GROTL5 GROTL6 GROTL6 GROTL6 GROTL6 GROTL6 GROTL6 GROTL7 GRO		
GPOLY.64 Group polynomial divide octlets GROTL.2 Group rotate left picks GROTL.4 Group rotate left picks GROTL.4 Group rotate left nibbles GROTL.6 GROTL.16 Group rotate left doublets GROTL.16 GROTL.32 Group rotate left doublets GROTL.32 Group rotate left doublets GROTL.32 Group rotate left cyadlets GROTL.128 Group rotate left pickets GROTL.128 GROTR.2 Group rotate left pickets GROTR.3 GROTR.3 GROTR.3 Group rotate right picks GROTR.8 GROTR.8 Group rotate right pickets GROTR.9 GROTR.9 GROTR.9 GROTR.16 GROTR.16 GROTR.16 GROTR.16 GROTR.16 GROTR.16 GROTR.17 GROTR.18 GROTR.18 Group rotate right doublets GROTR.18 GROTR.19 G		
GROTL2 GROYL 4 Group rotate left pecks GROTL8 Group rotate left hibbles GROTL8 Group rotate left hibbles GROTL18 Group rotate left hibbles GROTL18 Group rotate left doublets GROTL32 Group rotate left doublets GROTL64 Group rotate left vadlets GROTL64 Group rotate left hibbles GROTL64 Group rotate left hexlets GROTR128 Group rotate left hexlets GROTR2 Group rotate right pecks GROTR8 GROTR8 Group rotate right pibles GROTR8 Group rotate right obblets GROTR8 Group rotate right fobbles GROTR8 Group rotate right fobbles GROTR92 Group rotate right fobblets GROTR932 Group rotate right fobblets GROTR94 Group rotate right fobblets GROTR95 Group rotate right fobblets GROTR96 Group rotate right hexlets GROTR97 Group scatter pecks GSCATTER97 Group scatter ribbles GSCATTER98 Group scatter fobblets GSCATTER98 Group scatter doublets GSCATTER99 Group scatter fobbles GSCATTER99 Group shift left picks GSHL89 Group shift left picks GSHL89 Group shift left doublets		
GROTL 4 Group rotate left nibbles GROTL 18 Group rotate left nibbles GROTL 18 Group rotate left dyseles GROTL 18 Group rotate left dyseles GROTL 32 Group rotate left dyseles GROTL 34 Group rotate left dyseles GROTL 18 GROTL 18 GROTL 18 GROTL 18 GROTL 18 GROTL 18 GROTH 2 GROUP rotate left nextets GROTR 3 GROTR 3 GROTR 4 Group rotate right nibbles GROTR 8 GROTR 16 GROTR 16 GROTR 17 GROTT 18 GROTT 1		
GROTL 8 GROTL 16 GROTL 16 GROTL 16 GROTL 16 GROTL 22 GROTL 32 GROT		
GROTL 16 GROYL 32 GROYL 64 GRO		
GROTL-32 Group rotate left quadlets GROTL-64 Group rotate left octlets GROTL-64 Group rotate left hexlets GROTR-128 Group rotate left hexlets GROTR-128 Group rotate right pecks GROTR-14 Group rotate right ribbles GROTR-16 GROTR-16 Group rotate right oblets GROTR-16 GROTR-16 GROTR-16 GROTR-16 GROTR-16 GROTR-16 GROTR-16 GROTR-16 GROTR-17 GROTR-1		
GROTL 64 Group rotate left octiets GROTL 128 Group rotate left hexlets GROTH 2 Group rotate right pecks GROTH 3 GROTH 4 Group rotate right pecks GROTH 8 GROTH 9 GROTH		
GROTL 128 Group rotate left hexiets GROTR-2 Group rotate right pecks GROTR-4 Group rotate right pecks GROTR-4 Group rotate right pibbles GROTR-8 GROTR-16 GROTR-16 GROTR-16 GROTR-16 GROTR-16 GROTR-16 GROTR-17 GROTR-17 GROTR-17 GROTR-18 GR		
G.ROTR 2 Group rotate right pecks G.ROTR 4 Group rotate right ribbles G.ROTR 8 Group rotate right ribbles G.ROTR 16 G.ROTR 16 G.ROTR 16 G.ROTR 16 G.ROTR 17 G.ROTR 18		
GROTR 4 Group rotate right nibbles GROTR.16 Group rotate right bytes GROTR.16 Group rotate right doublets GROTR.32 Group rotate right doublets GROTR.32 Group rotate right quadlets GROTR.32 Group rotate right quadlets GROTR.42 Group rotate right cellets GROTR.128 Group rotate right cellets GROTR.128 Group scatter pecks GSCATTER.2 Group scatter pecks GSCATTER.4 Group scatter ribbles GSCATTER.8 Group scatter doublets GSCATTER.32 Group scatter doublets GSCATTER.32 Group scatter cellets GSCATTER.34 Group scatter doublets GSCATTER.35 GSCATTER.46 GROUP scatter doublets GSCATTER.46 GSCATTER.46 GROUP scatter fextlet GSCATTER.46 GROUP scatter fextlet GSCATTER.46 GROUP scatter flow file file file follows GSHL.4 Group shift left pocks GSHL.5 GROUP shift left doublets		
GROTR 8 Group rotate right bytes GROTR 16 GROTR 16 Group rotate right doublets GROTR 32 Group rotate right quadlets GROTR 64 GROUP G		
GROTR.16 Group rotate right doublets GROTR.32 Group rotate right doublets GROTR.32 Group rotate right quadelts GROTR.64 Group rotate right octlets GROTR.128 Group rotate right posters GSCATTER.2 Group scatter pecks GSCATTER.4 Group scatter ribbles GSCATTER.6 Group scatter oblets GSCATTER.16 Group scatter oblets GSCATTER.32 Group scatter quadelts GSCATTER.46 Group scatter quadelts GSCATTER.64 Group scatter coublets GSCATTER.64 Group scatter pudelts GSCATTER.64 Group scatter fextet GSCATTER.64 Group scatter fextet GSCATTER.64 Group scatter fextet GSCATTER.64 Group scatter fextet GSCATTER.64 Group shift left pecks GSHL.4 Group shift left pibels GSHL.6 Group shift left doublets		
GROTR 32 Group rotate right quadlets GROTR 64 Group rotate right quadlets GROTR 64 Group rotate right collets GROTR 64 GROUP rotate right hexlets GSCATTER.2 Group scatter pecks GSCATTER.4 Group scatter ribbles GSCATTER.8 GROUP scatter doublets GSCATTER.16 Group scatter doublets GSCATTER.32 Group scatter doublets GSCATTER.46 GSCATTER.64 GSCATTER.64 GSCATTER.64 GSCATTER.64 GSCATTER.64 GSCATTER.64 GSCATTER.64 GSCATTER.64 GSCATTER.65 GSCATTER.65 GSCATTER.65 GSCATTER.66 GSCATTER.66 GSCATTER.66 GSCATTER.66 GSCATTER.67 GSCATTER.67 GSCATTER.68 GSCATTER.68 GSCATTER.68 GSCATTER.69 GSCA		
G.ROTR.64 Group rotate right octlets G.ROTR.128 Group rotate right octlets G.SCATTER.2 Group scatter pecks G.SCATTER.4 Group scatter hibbles G.SCATTER.8 Group scatter bytes G.SCATTER.8 Group scatter doublets G.SCATTER.16 Group scatter doublets G.SCATTER.16 Group scatter doublets G.SCATTER.22 Group scatter doublets G.SCATTER.44 Group scatter doublets G.SCATTER.45 Group scatter fexter G.SCATTER.164 Group scatter hexter G.SCATTER.17842 Group scatter hexter G.SHL.1 Group shift left pecks G.SHL.4 Group shift left obblets G.SHL.16 Group shift left doublets		
GROTR-128 Group rotate right hexiets G.SCATTER.2 Group scatter pecks G.SCATTER.4 Group scatter nibbles G.SCATTER.8 Group scatter nibbles G.SCATTER.16 Group scatter obtate G.SCATTER.16 Group scatter doublets G.SCATTER.32 Group scatter doublets G.SCATTER.43 Group scatter coulsets G.SCATTER.64 Group scatter collets G.SCATTER.168 Group scatter hexiet G.SCATTER.1822 Group scatter hexiet G.SCATTER.1828 Group shift left pecks G.SHL.1 Group shift left pibles G.SHL.16 Group shift left doublets		
G SCATTER 2 Group scatter pecks G SCATTER 4 Group scatter hibbles G SCATTER.8 Group scatter bytes G SCATTER.18 Group scatter bytes G SCATTER.29 Group scatter quadlets G SCATTER.32 Group scatter quadlets G SCATTER.32 Group scatter relatel G SCATTER.12622 Group scatter relatel G SHL 2 Group shift left pecks G SHL 3 G GSHL 6 Group shift left bytes G SHL 16 Group shift left doublets		
G SCATTER.4 Group scatter nibbles G SCATTER.8 Group scatter bytes G SCATTER.16 Group scatter doublets G SCATTER.32 Group scatter doublets G SCATTER.432 Group scatter cettels G SCATTER.44 Group scatter cettels G SCATTER.45 Group scatter hexlet G SCATTER.128 <sup>22</sup> Group scatter hexlet G SHL.2 Group shift left pecks G SHL.4 Group shift left pibles G SHL.6 Group shift left doublets G SHL.6 Group shift left doublets		
G SCATTER 8 Group scatter bytes G SCATTER 16 Group scatter doublets G SCATTER.32 Group scatter quadlets G SCATTER.32 Group scatter quadlets G SCATTER.32 Group scatter related G SCATTER.12822 Group scatter related G SCATTER.12822 Group scatter related G SCATTER.12822 Group scatter in the scatter of the sca		
G SCATTER.16 Group scatter doublets G.SCATTER.32 Group scatter quadlets G.SCATTER.64 Group scatter cettes G.SCATTER.128*2 Group scatter cettes G.SCATTER.128*2 Group scatter hextet G.SHL.2 Group shift left pecks G.SHL.4 Group shift left nibbles G.SHL.6 Group shift left doublets G.SHL.16 Group shift left doublets		
G SCATTER.32 Group scatter quadlets G SCATTER.64 Group scatter octets G SCATTER.128 <sup>22</sup> Group scatter hexlet G.SHL.2 Group shift left pecks G.SHL.4 Group shift left nibbles G.SHL.6 Group shift left bytes G.SHL.6 Group shift left doublets		
G.SCATTER.64 Group scatter octlets G.SCATTER.12822 Group scatter hexlet G.SHL.2 Group shift left pecks G.SHL.4 Group shift left nibbles G.SHL.6 Group shift left bytes G.SHL.6 Group shift left bytes G.SHL.16 Group shift left bytes		
G.SCATTER.128 <sup>22</sup> Group scatter hextet G.SHL.2 Group shift left pecks G.SHL.4 Group shift left nibbles G.SHL.6 Group shift left bytes G.SHL.16 Group shift left doublets		
G.SHL.2 Group shift left pecks G.SHL.4 Group shift left nibbles G.SHL.8 Group shift left bytes G.SHL.16 Group shift left doublets		
G.SHL.4 Group shift left nibbles G.SHL.8 Group shift left bytes G.SHL.16 Group shift left doublets		
G.SHL.8 Group shift left bytes G.SHL.16 Group shift left doublets		
G.SHL.16 Group shift left doublets		
G.SHL.32 Group shift left quadlets		
	G.SHL.32	Group shift left quadlets

<sup>18</sup> C. NAND does not require a size specification, and is encoded as G.NAND.1.
19 G.NOR does not require a size specification, and is encoded as G.NOR.1.
20 C.OR does not require a size specification, and is encoded as G.NOR.1.
21 C.ORN does not require a size specification, and is encoded as G.OR.1.
21 C.ORN does not require a size specification, and is encoded as G.ORN.1. G.ORN is used as the encoding for SEFLUGE.1.
22 C.SCATTER 128 is encoded as G.SCATTER.1

G.SHL.64	Group shift left octlets
G.SHL. 128	Group shift left hexlets
G.SHR.2	Group signed shift right pecks
G.SHR.4	Group signed shift right nibbles
G.SHR.8	Group signed shift right bytes
G.SHR.16	Group signed shift right doublets
G.SHR.32	Group signed shift right quadlets
G.SHR.64	Group signed shift right octlets
G.SHR.128	Group signed shift right hexlets
G.U.DIV.64	Group signed divide octlets
G.U.EXPAND.1	Group unsigned expand bits
G.U.EXPAND.2	Group unsigned expand pecks
G.U.EXPAND.4	Group unsigned expand nibbles
G.U.EXPAND.8	Group unsigned expand bytes
G.U.EXPAND.16	Group unsigned expand doublets
G.U.EXPAND.32	Group unsigned expand quadlets
G.U.EXPAND.64	Group unsigned expand octlet
G.U.MUL.2	Group unsigned multiply pecks
G.U.MUL.4	Group unsigned multiply nibbles
G.U.MUL.8	Group unsigned multiply bytes
G.U.MUL.16	Group unsigned multiply doublets
G.U.MUL.32	Group unsigned multiply quadlets
G.U.MUL.64	Group unsigned multiply octlets
G.U.SHR.2	Group unsigned shift right pecks
G.U.SHR.4	Group unsigned shift right nibbles
G.U.SHR.8	Group unsigned shift right bytes
G.U.SHR.16	Group unsigned shift right doublets
G.U.SHR.32	Group unsigned shift right quadlets
G.U.SHR.64	Group unsigned shift right octlets
G.U.SHR.128	Group unsigned shift right hexlets
G.XNOR <sup>23</sup>	Group exclusive-nor
G.XOR <sup>24</sup>	Group exclusive-or

<sup>&</sup>lt;sup>2)</sup>G.XNOR does not require a size specification, and is encoded as G.XNOR.1. G.XNOR is used as the encoding for G.SET.E.1. <sup>24</sup>G.XOR does not require a size specification, and is encoded as G.XOR.1. G.XOR is used as the encoding for G.ADD.1, G.SUB.1 and G.SET.N.E.1.

class	op	size
linear	ADD	2 4 8 16 32 64
bitwise	AND ANDN NAND NOR OR ORN XNOR XOR	
signed multiply	MUL	1 2 4 8 16 32 64
unsigned multiply	U.MUL	2 4 8 16 32 64
signed divide	DIV	64
unsigned divide	U.DIV	64
	GATHER SCATTER	2 4 8 16 32 64
galois field	POLY	1 2 4 8 16 32 64
precision	COMPRESS EXPAND U.EXPAND	1 2 4 8 16 32 64
shift ·	ROTR ROTL SHR SHL U.SHR	2 4 8 16 32 64 128

#### Format

G.op.size

rc=ra.rb

31	24	23 18	17 12	11 6	5 0
G.size	9	ra	rb	rc	op
8		6	6	6	6

#### Description

Two values are taken from the contents of registers or register pairs specified by ra and rb. The specified operation is performed, and the result is placed in the register or register pair specified by rc.

A reserved instruction exception occurs if rcg is set, and for certain operations, if ran or rbn is set.

#### Definition

def Group(op,size,ra,rb,rc)

case op of
G.MUL, G.U.MUL, G.DIV, G.U.DIV:
a ← RegRead(ra, 64)
b ← RegRead(rb, 64)

G.ADD, G.SUB, G.SET.L, G.SET.UL, G.SET.E, G.SET.NE, G.SET.GE, G.SET.UGE, G.AND, G.OR, G.XOR, G.ANDN, G.NAND, G.NOR, G.XNOR, G.ORN,

G.GATHER, G.SCATTER:

G.GAIRER, G.SOVALLER: a ← RegRead(ra, 128) b ← RegRead(rb, 128) G.COMPRESS, G.ROTL, G.ROTR, G.SHL, G.SHR, G.U.SHR, G.POLY: a ← RegRead(ra, 128) b Pachandric (ch)

b ← RegRead(rb, 64)

G.EXPAND, G.U.EXPAND:

a ← RegRead(ra, 64)

```
b ← RegRead(:b, 64)
endcase
case op of
      G.ADD:
            for i ← 0 to 128-size by size
                  Ci+size-1..i ← ai+size-1..i + bi+size-1..i
            endfor
      G.MUL:
            for i ← 0 to 64-size by size
                 C2*(i+size)-1..2*i ← (asize-1 size || asize-1+i..i) * (bsize-1 size || bsize-1+l..i)
      G.U.MUL:
            for i ← 0 to 64-size by size

C2*(i+size)-1..2*i ← (0size || a<sub>size-1+i..i</sub>) * (0size || b<sub>size-1+i..i</sub>)
      G.DIV:
            if (b = 0) or ( (a = (111063)) and (b = 164) ) then
                 c ← undefined
            else
                  q \leftarrow a/b
                  r ← a · o b
                  c ← r63..0 ll q63..0
            endif
     G.U.DIV:
            if b = 0 then
                 c ← undefined
            else
                  q \leftarrow (0 \parallel a) / (0 \parallel b)
                  r ← a - q b
                  c ← r<sub>63..0</sub> II q<sub>63..0</sub>
            endif
     G.AND:
     c ← a and b
G.OR:
           c ← a or b
     G.XOR:
            c ← a xor b:
     G.ANDN:
           c ← a and not b
     G.NAND:
     c ← not (a and b)
G.NOR:
     c ← not (a or b)
G.XNOR:
           c ← not (a xor b)
     G.ORN:
           c ← a or not b
     G.POLY:
           p[0] \leftarrow a
for i \leftarrow 1 to size
                 p[i] \leftarrow (p[i-1]_0?(0^{64} \parallel b): 0^{128}) \text{ xor } (p[i-1]_0 \parallel p[i-1]_{127..1})
           endfor
     c ← p[size]
G.GATHER;
           for k ← 0 to 128-size by size
                 for i ← k to k+size-1 by 1
```

```
if a<sub>i</sub> then
                         c<sub>i</sub> ← b<sub>i</sub>
                   j ← j + 1
endif
             endfor
             j ← k+size-1
             for I ← k+size-1 to k by -1
                   if ~a; then
                         c_i \leftarrow b_i
                  j ← j - 1
endif
             endfor
endfor
G.SCATTER:
      for k ← 0 to 128-size by size
            j ← k
             for i ← k to k+size-1 by 1
                   if a<sub>i</sub> then
                         ci← bi
                  j \leftarrow j + 1 endif
             endfor
             i ← k+size-1
             for i ← k+size-1 to k by -1
                  if ~a; then
                         ci ← bi
                  j ← j - 1
endif
            endfor
      endlor
G.COMPRESS:
      for i ← 0 to 64-size by size
            Ci+size-1..i ← ai+i+size-1+(b&(size-1))..i+i+(b&(size-1))
      endfor
G.EXPAND:
      for i ← 0 to 64-size by size
            c_{i+1+size+size+1...i+i} \leftarrow a_{i+size-1}^{size-(b\&(size-1))} \parallel a_{i+size-1,i} \parallel 0^{b\&(size-1)}
G.U.EXPAND:
      for i ← 0 to 64-size by size
            Ci+i+size+size-1..i+i ← 0<sup>size</sup>-(b&(size-1)) || a; size-1.i || 0<sup>b&(size-1)</sup>
      endfor
G.ROTL:
      for i ← 0 to 128-size by size
            Ci+size-1..i ← ai+size-1-(b&(size-1))..i li rii+size-1.i+size-1-(b&(size-1))
      endfor
G.ROTR:
      for i ← 0 to 128-size by size
            Ci+size-1..i ← ai+(b&(size-1))-1..i | 8:+size-1..i+(b&(size-1))
      endfor
G.SHL:
      for i ← 0 to 128-size by size
           Ci+size-1..i ← ai+size-1-(b&(size-1))..i || 0b&(size-1)
      endfor
G.SHR:
```

Ior i ← 0 to 128-size by size

Ci+siz+1.i ← 8i+size-1b8(size-1)d ai+size-1.i+(b8(size-1))

endfor
G.U.SHR:
Ior i ← 0 to 128-size by size

Ci+size-1.i ← 0b8(size-1)d ai+size-1.i+(b8(size-1))

endfor
end case

op of

G.AD. C. G.M.L. G.UMUL, G.DIV, G.UDIV:

G.AD. C. G.R. G.XOR, G.ANDN, G.NAND, G.NOR, G.XNOR, G.ORN,

G.PARD O LEPARN, G.SHL, G.SHR, G.U.SHR,

G.GATHER, G.SCATTER, G.POLY:

G.COMPRESS:

G.GOMPRESS:

RegWitte(c), 28, c)

endcase enddef

Exceptions

Reserved Instruction

- 109 -